**逻辑事件派发器v0.1**

**研发八部**

模块名：<weedong/core/event/event.h>

依赖: <weedong/core/os.h>

示例：

#include "os.h"

#include "event/event.h"

using namespace wd;

enum

{

eEvent\_UseItem,

eEvent\_CompleteTask,

eEvent\_Max,

};

struct Event\_UseItem: public CEventMgr::EventBase

{

Event\_UseItem()

{

nEventID = eEvent\_UseItem;

nItemID = 0;

}

uint32 nItemID;

};

struct Event\_CompleteTask: public CEventMgr::EventBase

{

Event\_CompleteTask()

{

nEventID = eEvent\_CompleteTask;

nTaskID = 0;

}

uint32 nTaskID;

};

void OnUseItem(CEventMgr::EventBase\* pEvent)

{

Event\_UseItem\* pInfo = (Event\_UseItem\*)pEvent;

pInfo = NULL;

}

void OnCompleteTask(CEventMgr::EventBase\* pEvent)

{

}

int main()

{

CEventMgr eventMgr(eEvent\_Max);

int nItemRegID = eventMgr.Register(eEvent\_UseItem, OnUseItem);

int nItemRegID2 = eventMgr.Register(eEvent\_UseItem, OnUseItem);

int nTaskRegID = eventMgr.Register(eEvent\_CompleteTask, OnCompleteTask);

Event\_UseItem iteminfo;

iteminfo.nItemID = 1234;

eventMgr.Dispatch(eEvent\_UseItem, &iteminfo);

Event\_CompleteTask taskinfo;

taskinfo.nTaskID = 1234;

eventMgr.Dispatch(eEvent\_CompleteTask, &taskinfo);

eventMgr.UnRegister(nItemRegID);

eventMgr.UnRegister(nItemRegID2);

eventMgr.UnRegister(nTaskRegID);

return 0;

}